TdaExecutive also creates a public singleton instance of Exic. Exic is the wrapper object for an EXIC parameter file. In the Transducer Design Advisor, all state information which is in response to user inputs and which must persist from window to window, and across program invocations, is stored in the Exic object. When the user presses the Next button in a window, TdaFrame calls the getSuccessor() method of the singleton WindowNavigator object, which is also created by TdaExecutive. This method examines various public parameters in the Exic object to decide which of the (possibly several) successor windows to the current window should be chosen, and returns the name of the chosen window to TdaFrame. The TdaFrame then calls the changeToWindow() method of TdaExecutive.

[0063] A number of the windows in the Transducer Design Advisor allow the user to use the mouse to drag graphical objects in order to set some value. One example is in the wCTQWeights window (shown in FIG. 11), where the user can specify the relative weights for different CTQs by dragging nodes in a profile curve up and down. Another example is in the azimuthal simulation set-up windows. Here, the user can drag a red line indicating the beam position (for linear probes) or beam angle (for curved and sector probes). An alphanumeric readout shows the numerical value of the beam angle or position and is continually updated as the user drags the line around. If the user types a new value into the readout text field, the line is instantly updated to reflect the new value. In addition to the beam angle or position, the user can also drag the corners of a green rectangle (for linear probes) or a wedge (for curved or sector probes) to specify the minimum and maximum azimuth, and the minimum and maximum range for sampling. As with the beam, there are textual readouts that show the numerical values of these parameters, and if the text fields are updated, the graphical display is also updated.

[0064] The "sensitive" areas that will respond to mouse clicks and drags are shown on the screen. In accordance with the preferred

embodiment, small square boxes are used to show the user where to click. As the user drags these boxes, their motion is constrained so that they can only move into locations that "make sense". For instance, for a curved probe, the sensitive region for specifying the beam angle is constrained to move in a circular arc.

[0065] To support this kind of mouse-driven animation and to support useful features such as constraining the motion to a particular set of loci, a set of classes have been designed that, when subclassed, implement the animation features. Referring to FIG. 9, the DragablePanel class is a subclass of JPanel. It implements the basic "canvas" in which various dragable sensitive areas can live. A DragablePanel can support any number of sensitive areas; internally, they are organized in a Linked List. The HotSpot class is the superclass for all sensitive areas. It is basically a rectangle, which may or may not have a visual representation. A HotSpot is mainly a passive entity: the methods that cause it to be sensitive to mouse activity are in DragablePanel. A DragModel is attached to each HotSpot. The DragModel defines how the HotSpot responds to mouse motion: in particular, the DragModel implements the constraints on how (and if) the HotSpot can move in response to dragging by the mouse. The DragablePanel, HotSpot, and DragModel together are an example of the well-known Model-View-Controller object pattern.

[0066] FIG. 9 shows a typical instance of usage for these classes. The wSimCurvedAzimuth window is where the user specifies the azimuthal and range limits for a simulation. The CurvedScan object is a subclass of CurvedDragablePanel, which is itself a subclass of DragablePanel. CurvedScan owns three HotSpots: one for the beam position and two for the upper-right and lower-left corners of a sector that specifies the azimuth and range limits. Each HotSpot is associated with a respective subclass of DragModel that specifies how that HotSpot can move as the user drags the mouse.

[0067] In general, regardless of the type of transducer the user wants to simulate, the user's journey through the Transducer Design Advisor will take a similar path (shown in FIG. 7). The user starts by telling the Transducer Design Advisor whether the user wants to edit an existing set of files or whether the user is creating a new set from scratch. In either case, the user will use a file browser window interface to pick a folder that contains the files, and to pick a root name for the file set. Next, the user specifies various geometric characteristics of the transducer, such as the number of layers in each transducer element (see FIG. 5), which multi-row technology to use (see FIG. 6), the number of rows of elements, the transducer's elevational and azimuthal size, the number of elements per row, and so on. Next, the user specifies how the user wants to simulate the imager system, picking parameters such as the major system mode to simulate (BMode, Color Flow, or PE Doppler), the azimuthal and elevation planes that the user wants to sample in the simulation, the excitation waveform to use, and so on. Finally, the user specifies weights for the various CTQs at different depths (see FIG. 11). These weights will be included in the aforementioned CTQ Policy file.

[0068] In the window for opening existing or creating new files (indicated by the block 27 in FIG. 7), the user must pick one of three choices: open an existing file set but save the output files under a different name; open an existing file set to edit its content in place; or create a new file set from scratch. Regardless of which one of these options the user picks, when the user clicks the Next button, the user will be presented with a standard file browser dialog box which the user uses to specify a folder and a root name for the new or existing file set. For the first two options, the user will browse to an existing file set and choose any of the files in that set. For the third option, the user browses to some folder where the user wants the output files to be written, and then enters the user's choice for the root name for the new file set. If the user chooses the first option, the user will actually be presented with two file browser dialog boxes: one to specify the existing file set and one to specify the new name for the output files.